Session 114

Application Packaging and Document Binding

Christopher Linn
User Experience Engineering

Application Packaging Overview

- Application evolution
- Packages and bundles
- Application package structure
- Property lists
- Document types
- URL schemes

Document Binding Overview

- What is document binding?
- What are the binding rules?
- Developer guidelines
- Launch Services API
- The bindings database

What Is an Application?

What Is an Application?

Executable Code

What Is an Application?

Executable Code

Strings

What Is an Application?

Executable Code

Strings

Layouts

What Is an Application?

Executable Code

Icons

Strings

Layouts

What Is an Application?

Executable Code

Icons

Strings

Layouts

Help Files

What Is an Application?

Executable Code

Icons

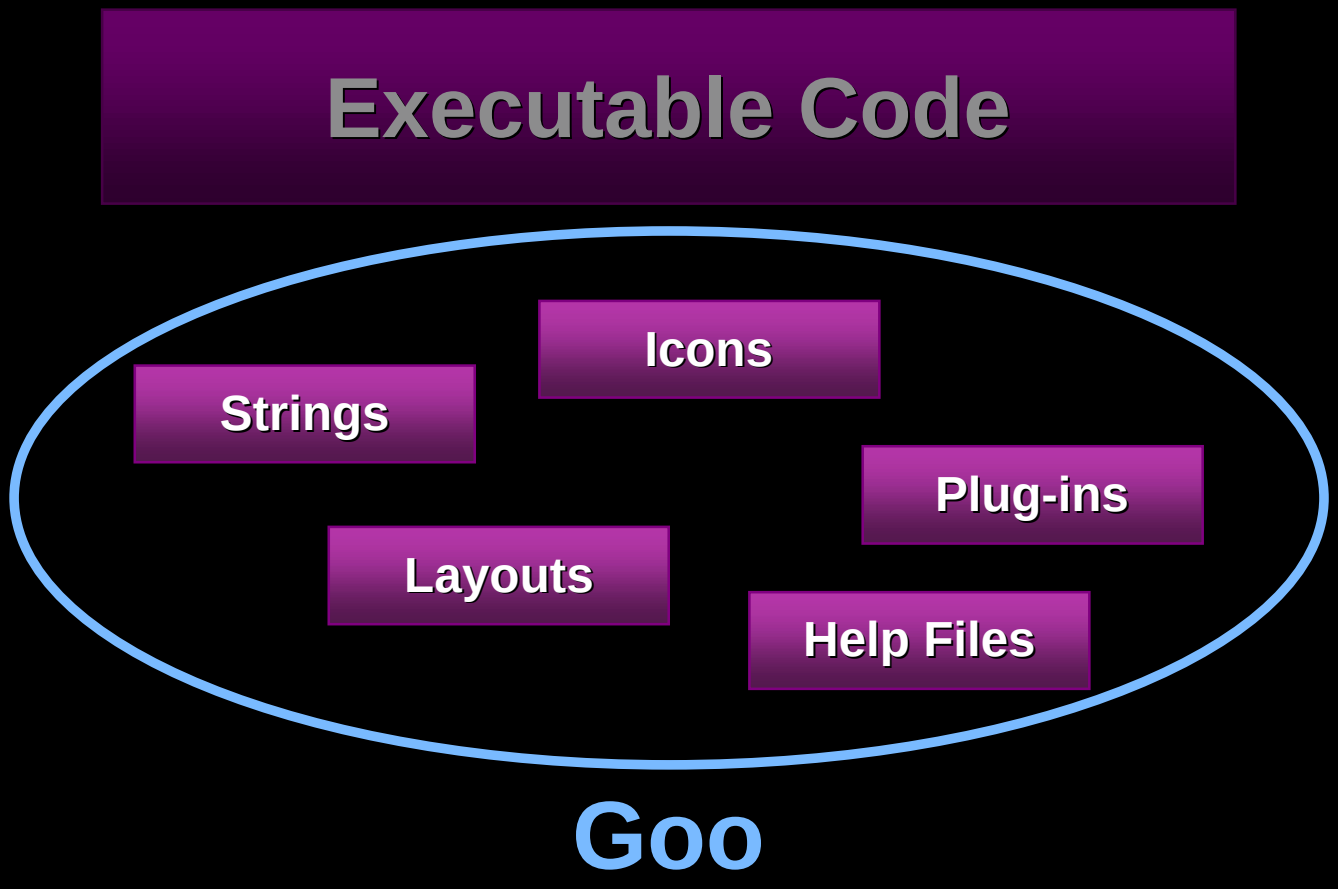
Strings

Plug-ins

Layouts

Help Files

What Is an Application?



The Single File Solution

- The goo is hidden in the resource fork
- Resources are easily localized
- Fully supported for native applications on Mac OS X

The Single File Solution

Limitations

- Contains only one localization
- Not flexible or extensible
- No place to put help files, plug-ins, other goo

Application Packages

A better solution

- Everything goes into a directory
- The user sees it as a file
- All the goo is hidden
- Flexible and extensible

Application Packages

A better solution

- Multiple binaries
- Multiple localizations
- Supports both CFM and Mach-O binaries
- Can be launched on Mac OS 9
- Not just for applications

Bundle vs. Package

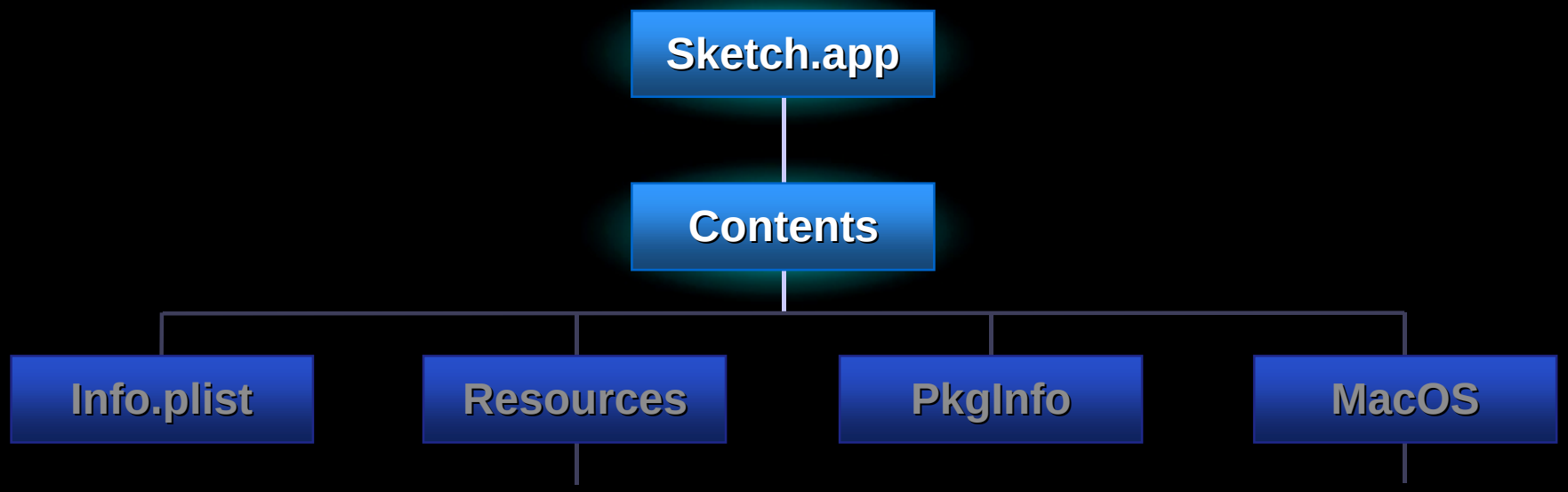
Bundle

A directory containing code + resources

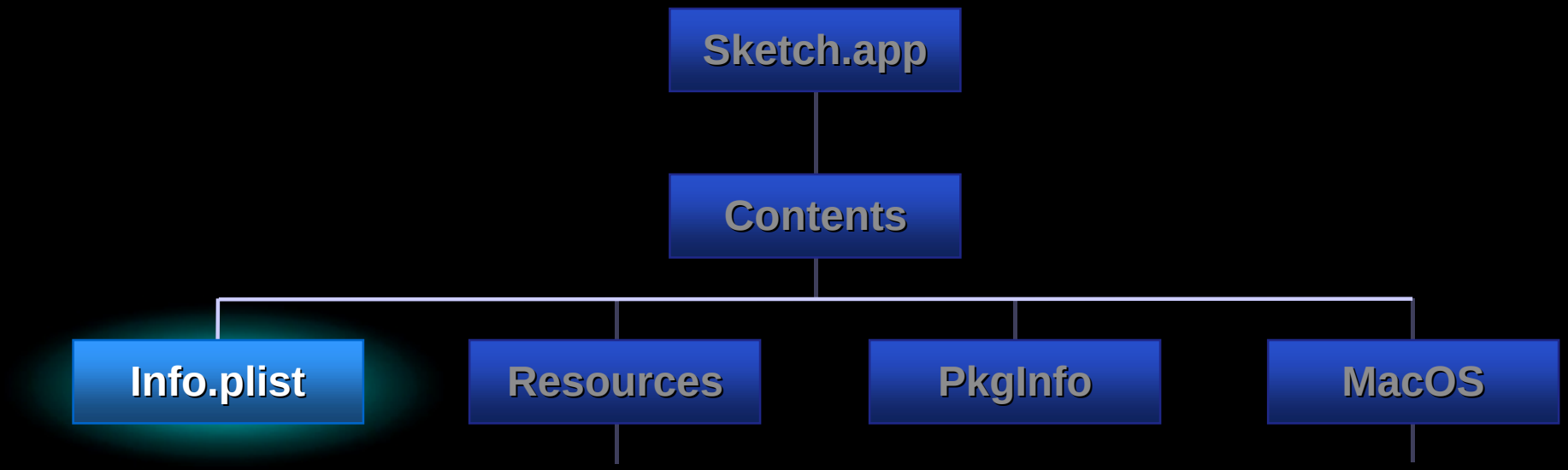
Package

A directory presented as a file

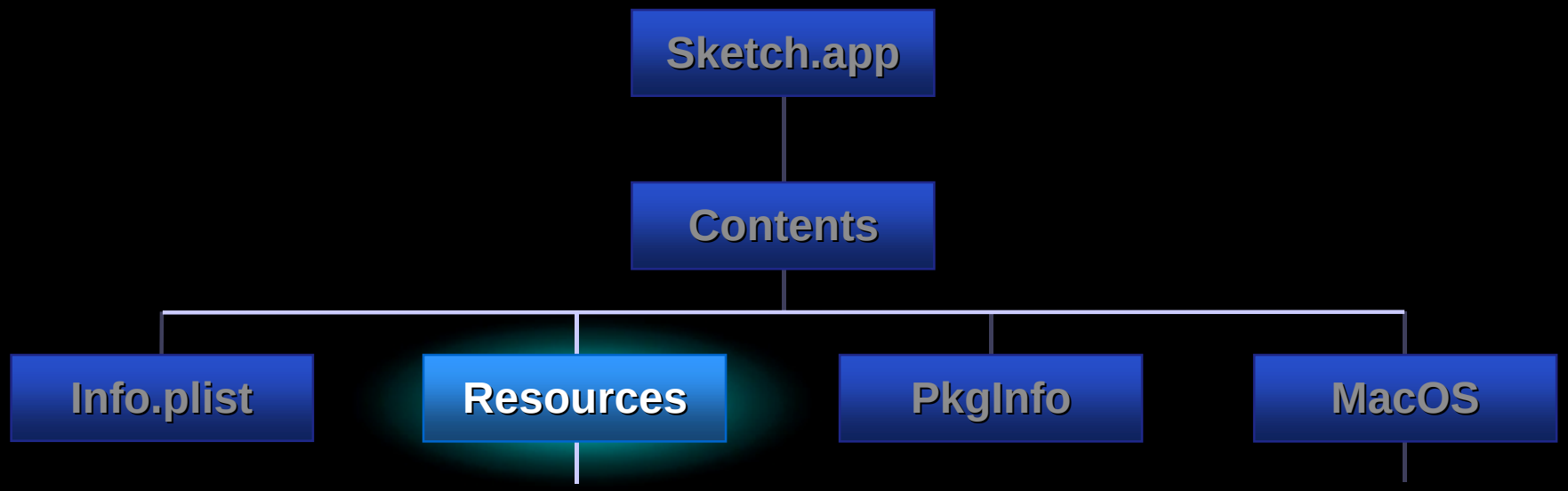
Application Package Structure



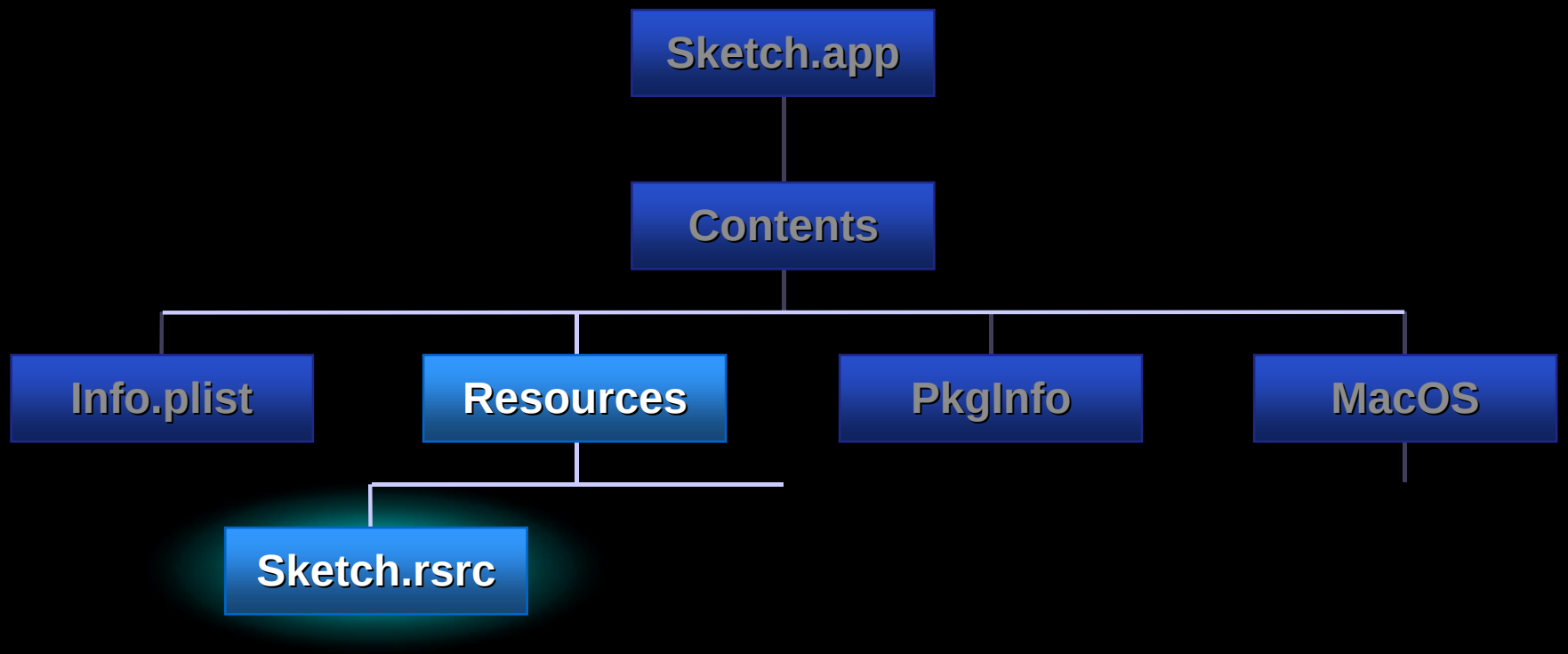
The Property List



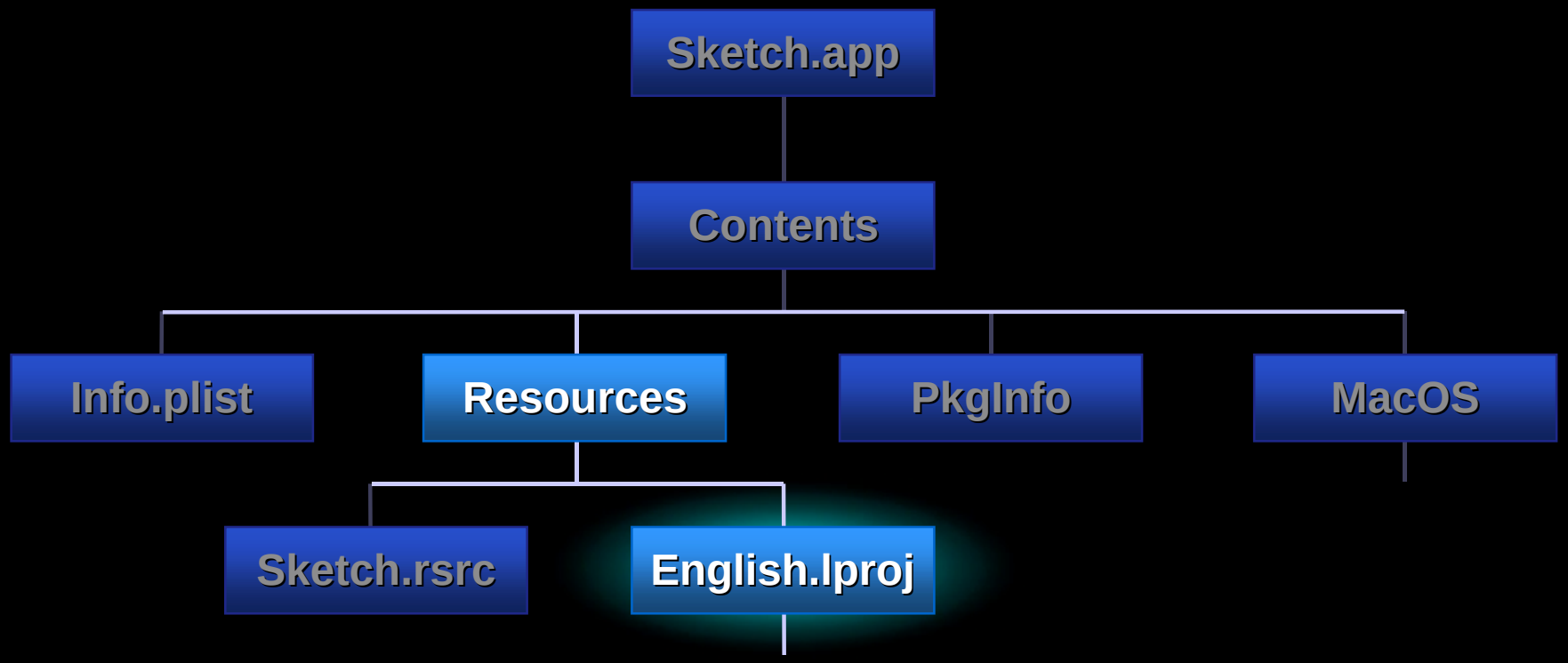
Resources Directory



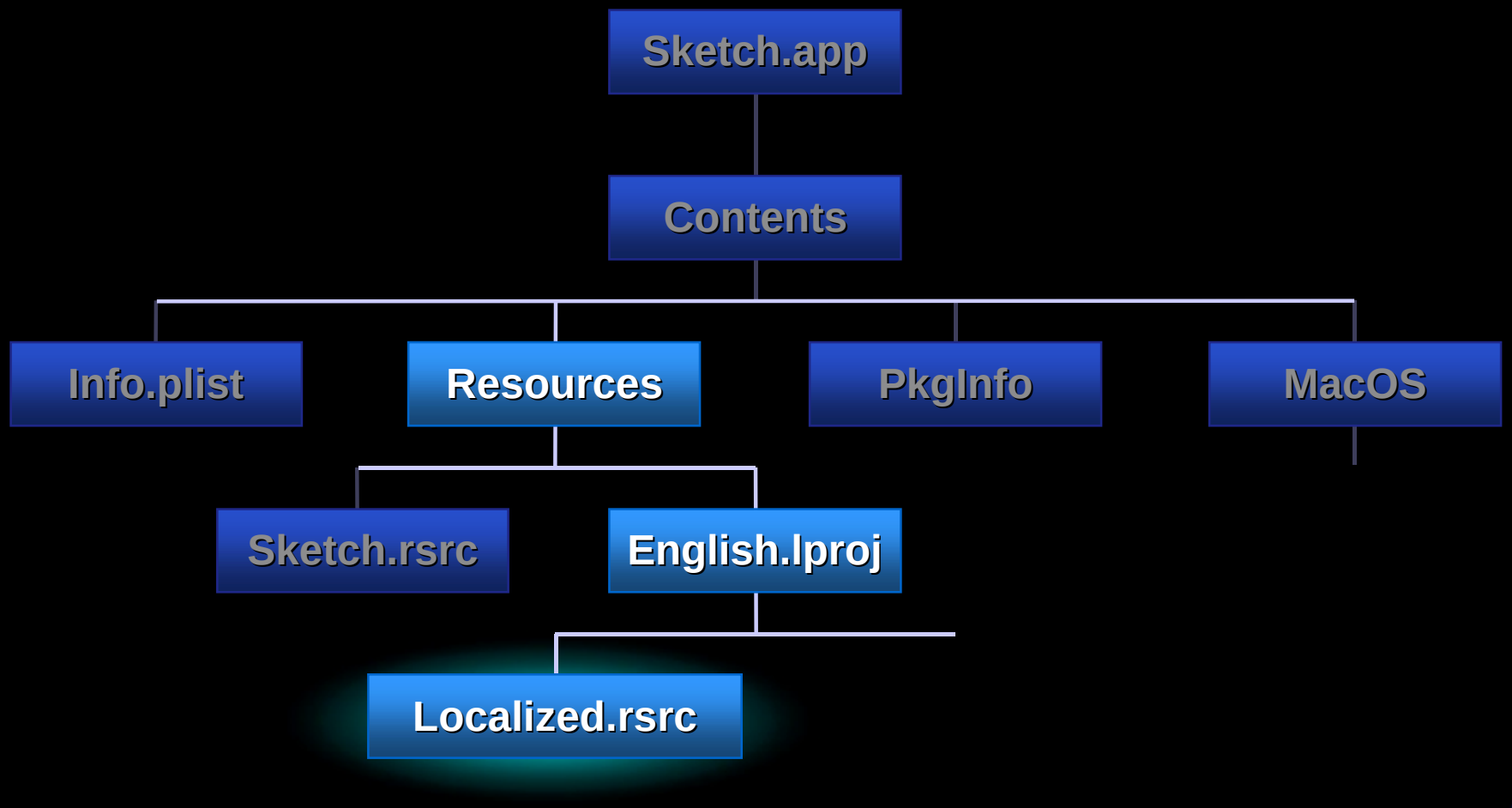
Global Resource Files



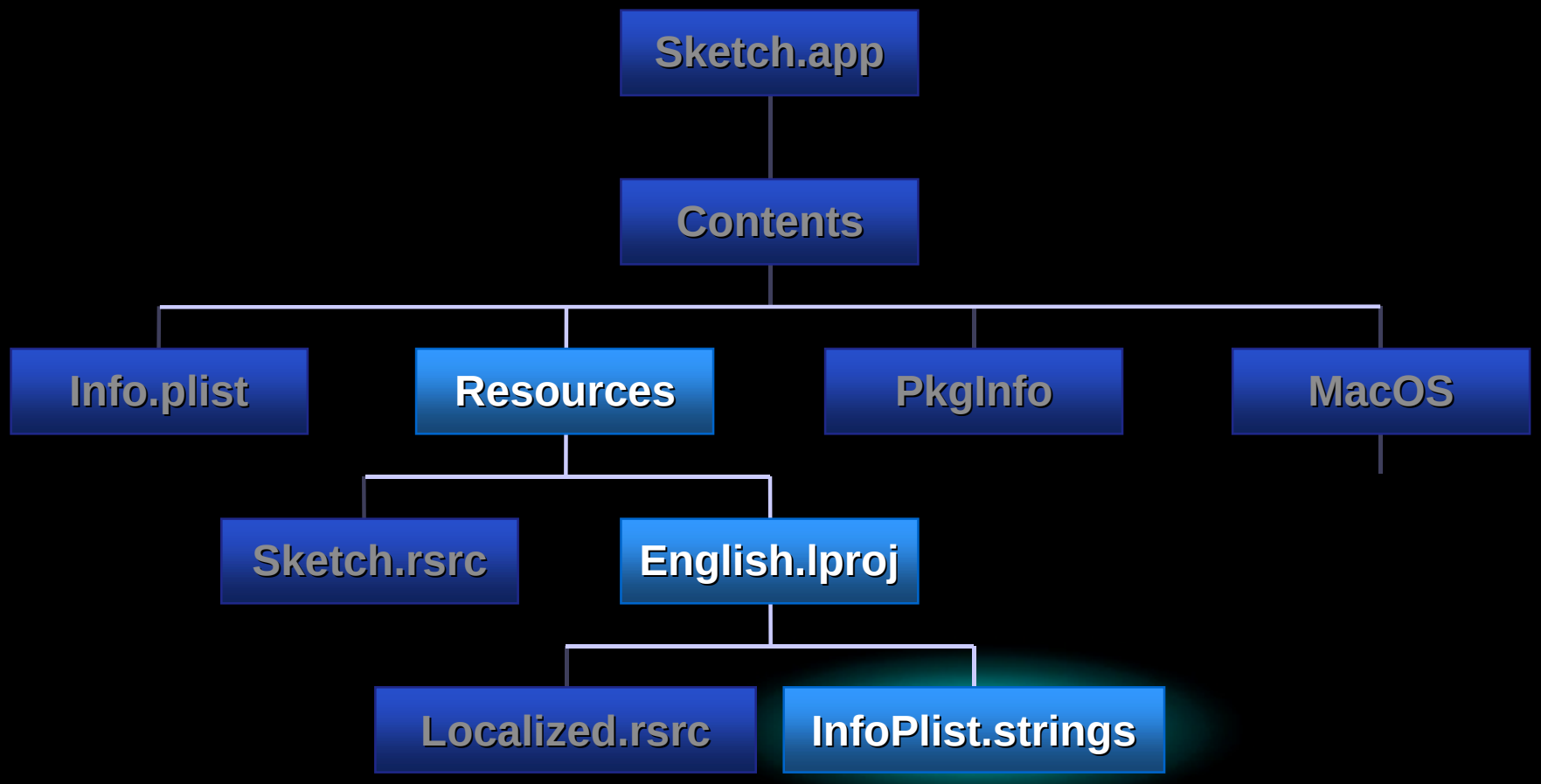
Localized Resource Directories



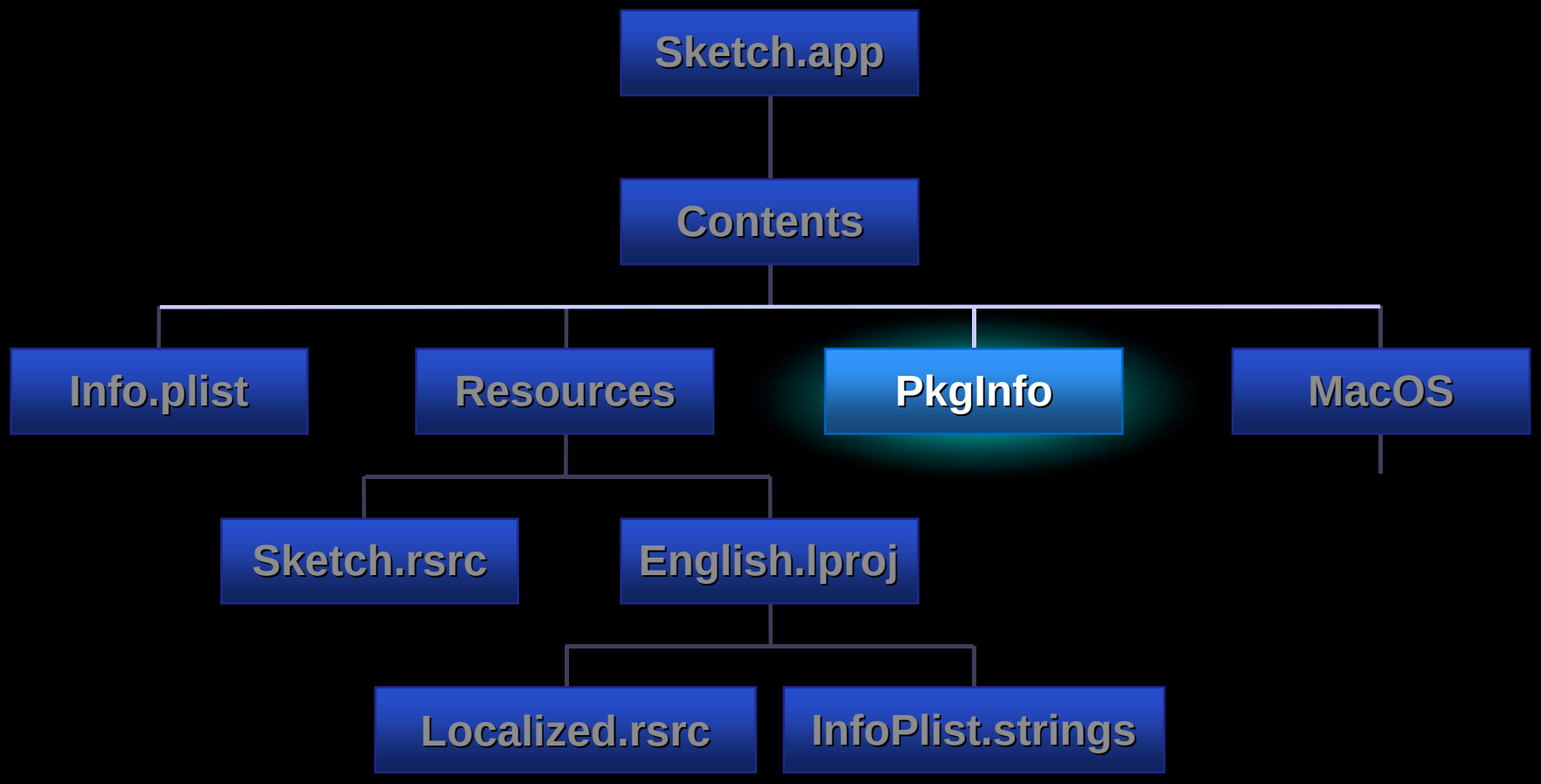
Localized Resource Files



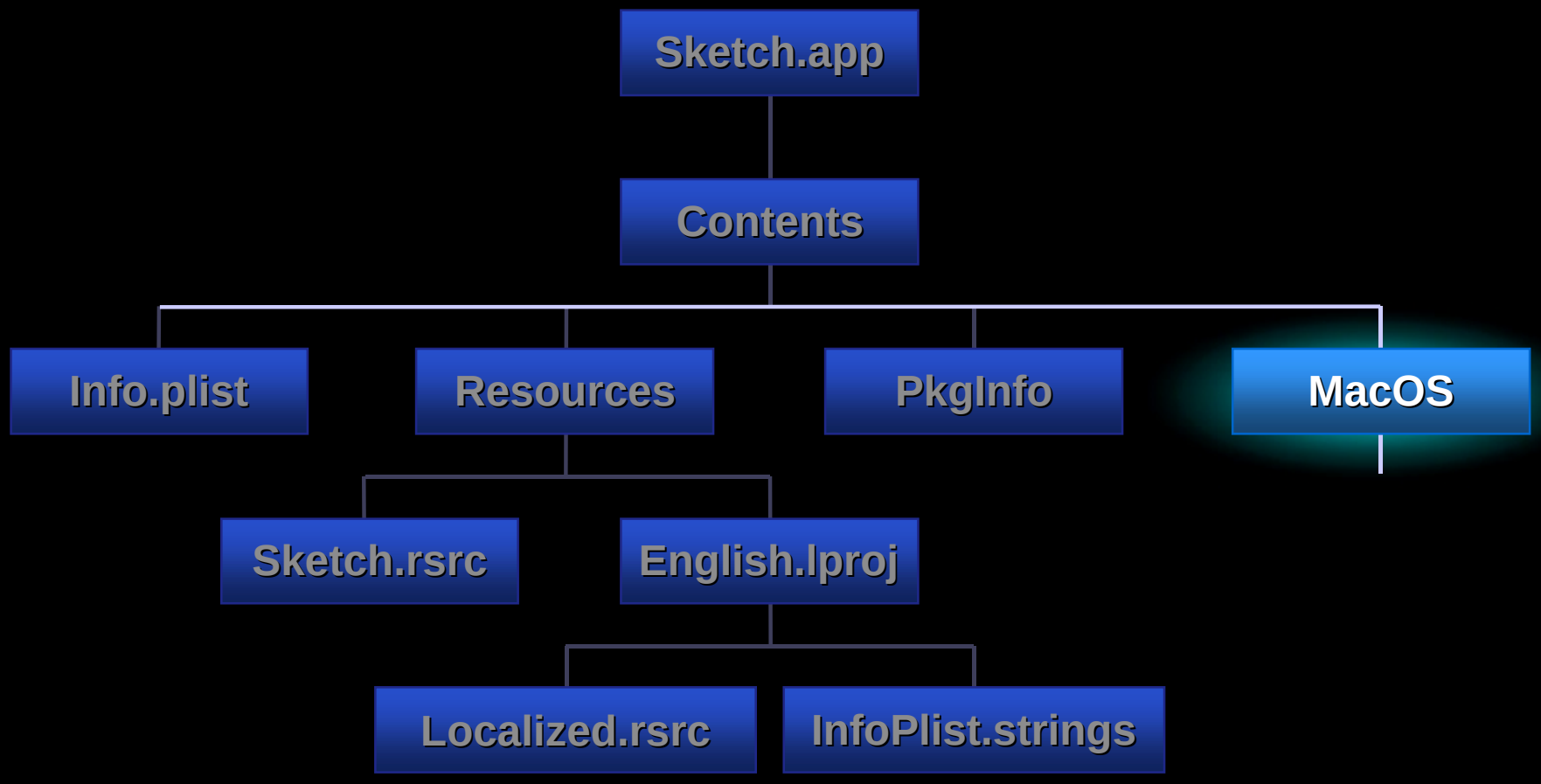
Localized Property List Strings



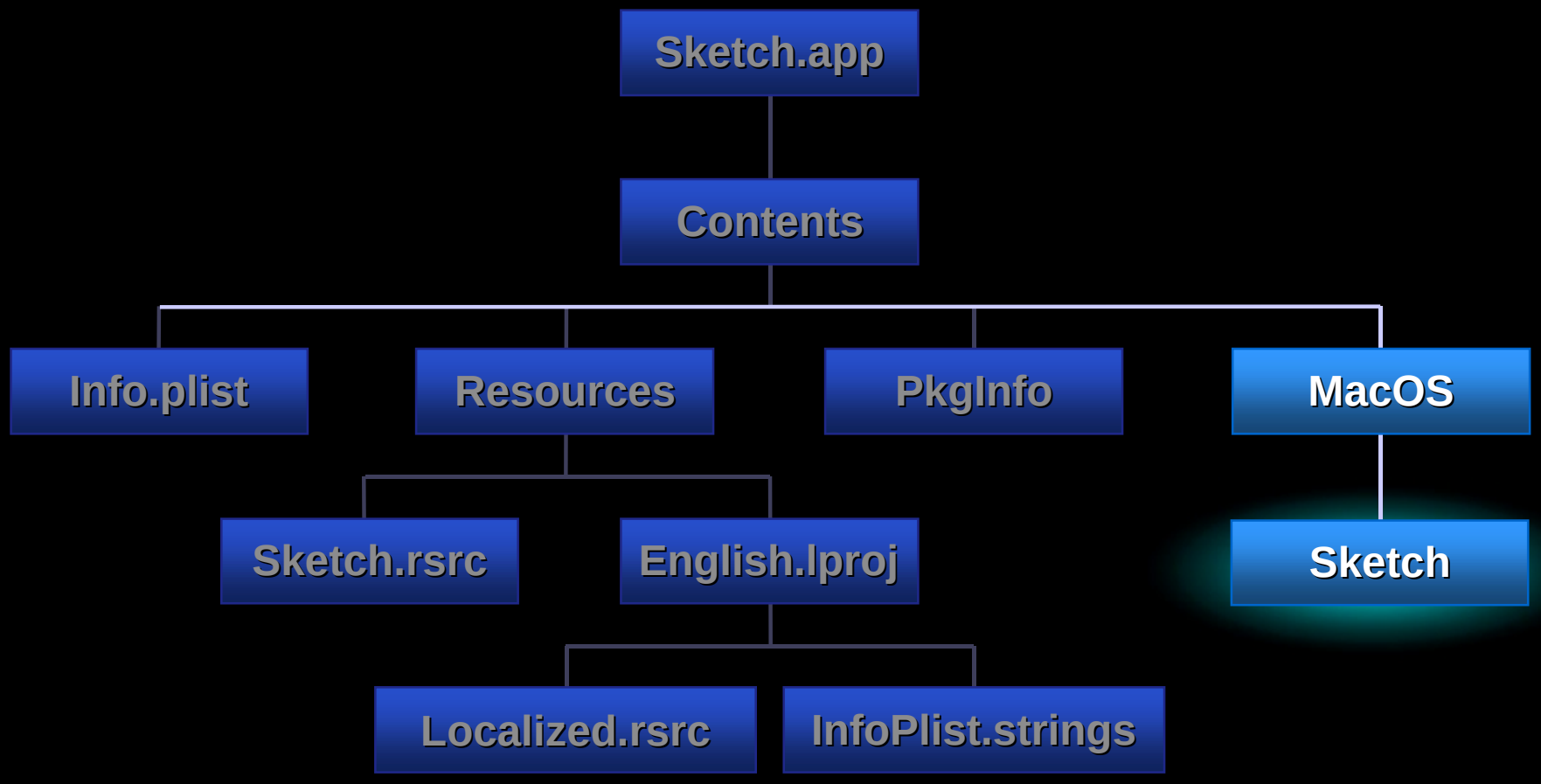
The PkgInfo File

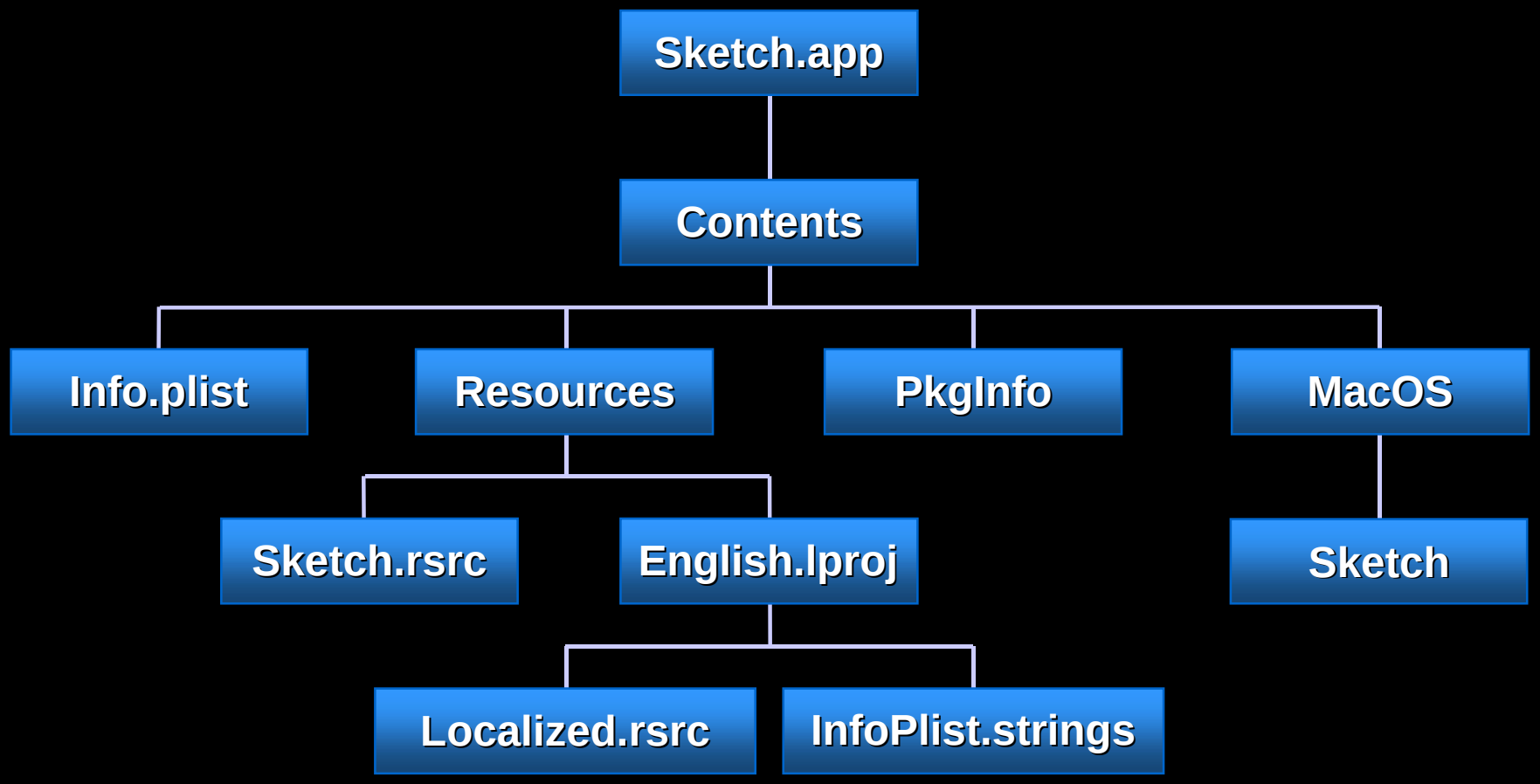


Executable Directories

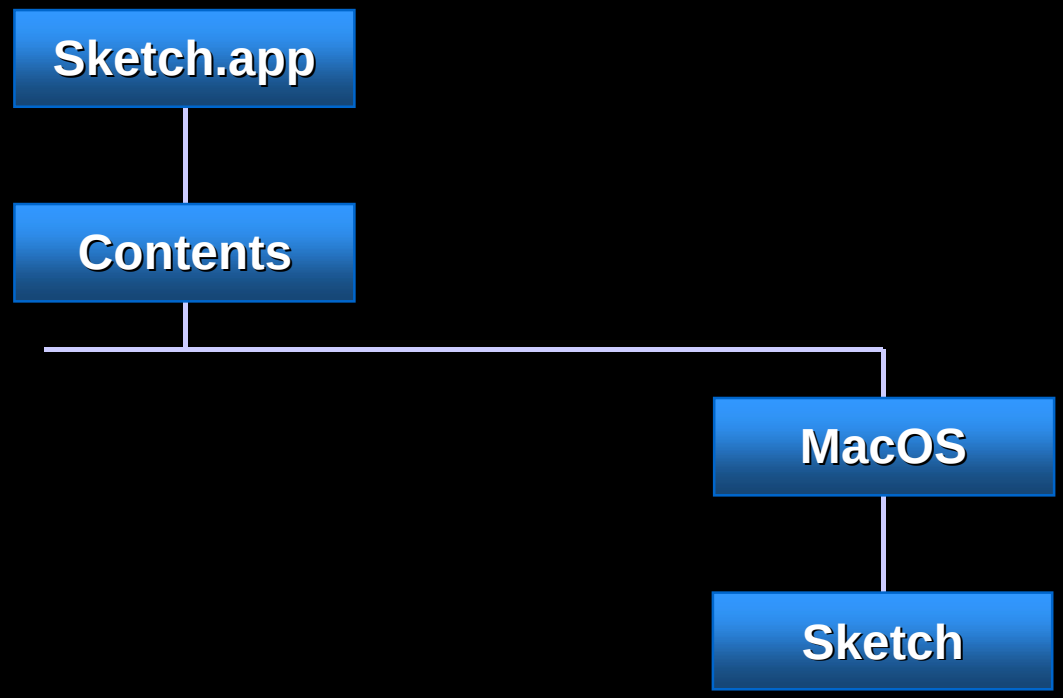


The Executable

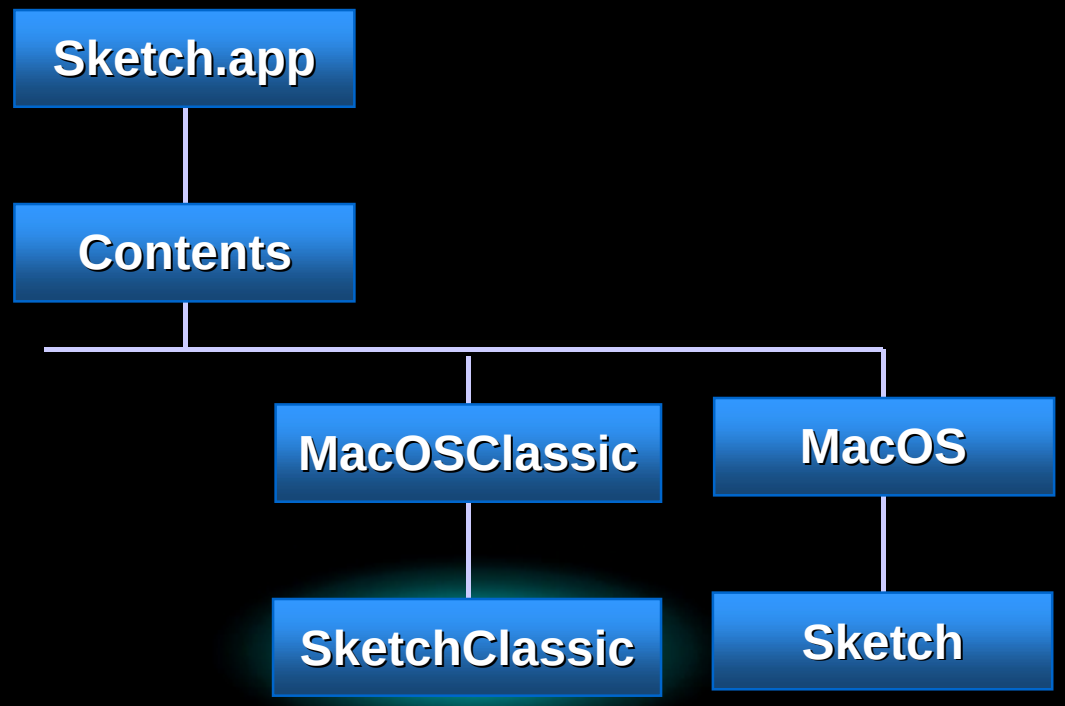



Sketch.app
Contents
Info.plist
Resources
PkgInfo
MacOS
Sketch.rsrc
English.lproj
Sketch
Localized.rsrc
InfoPlist.strings

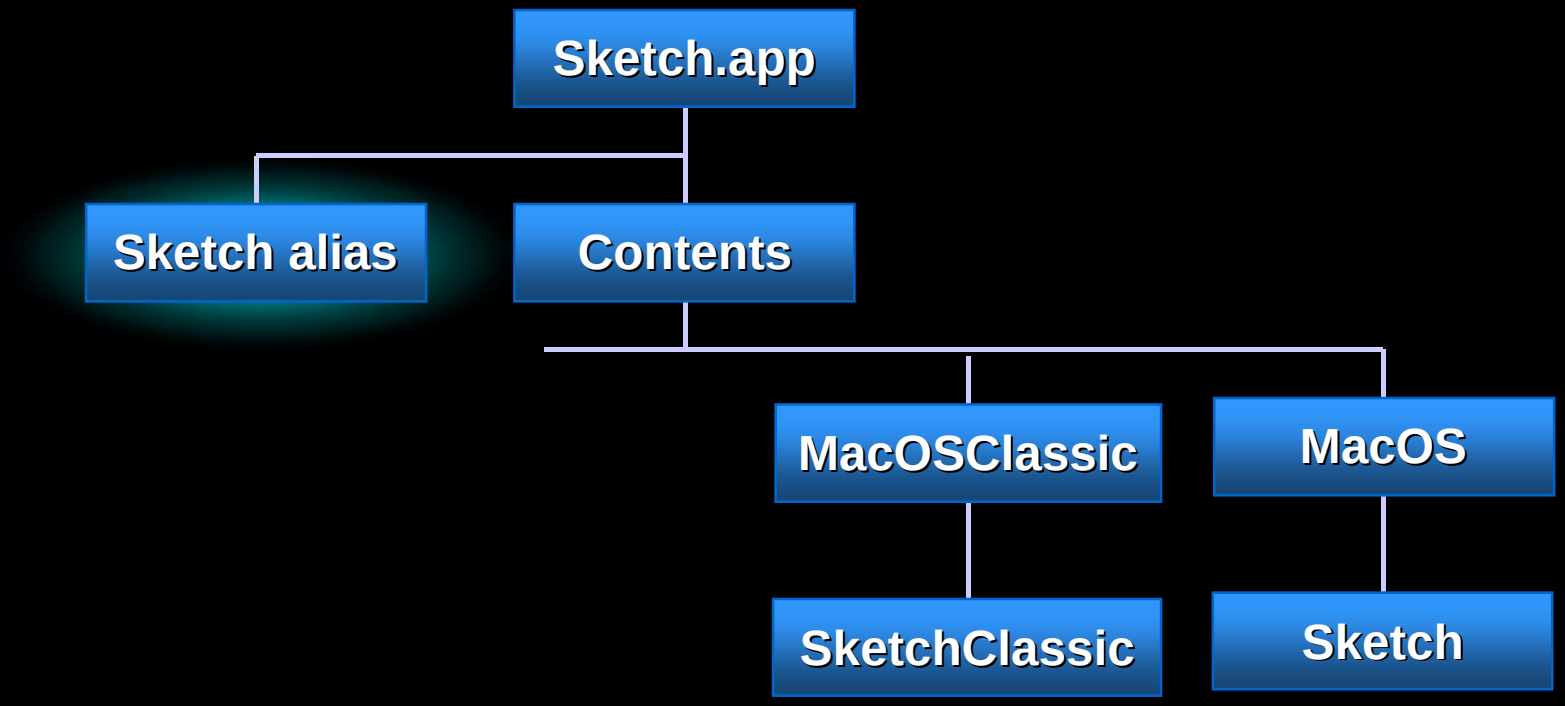
Application Packages for Mac OS 9



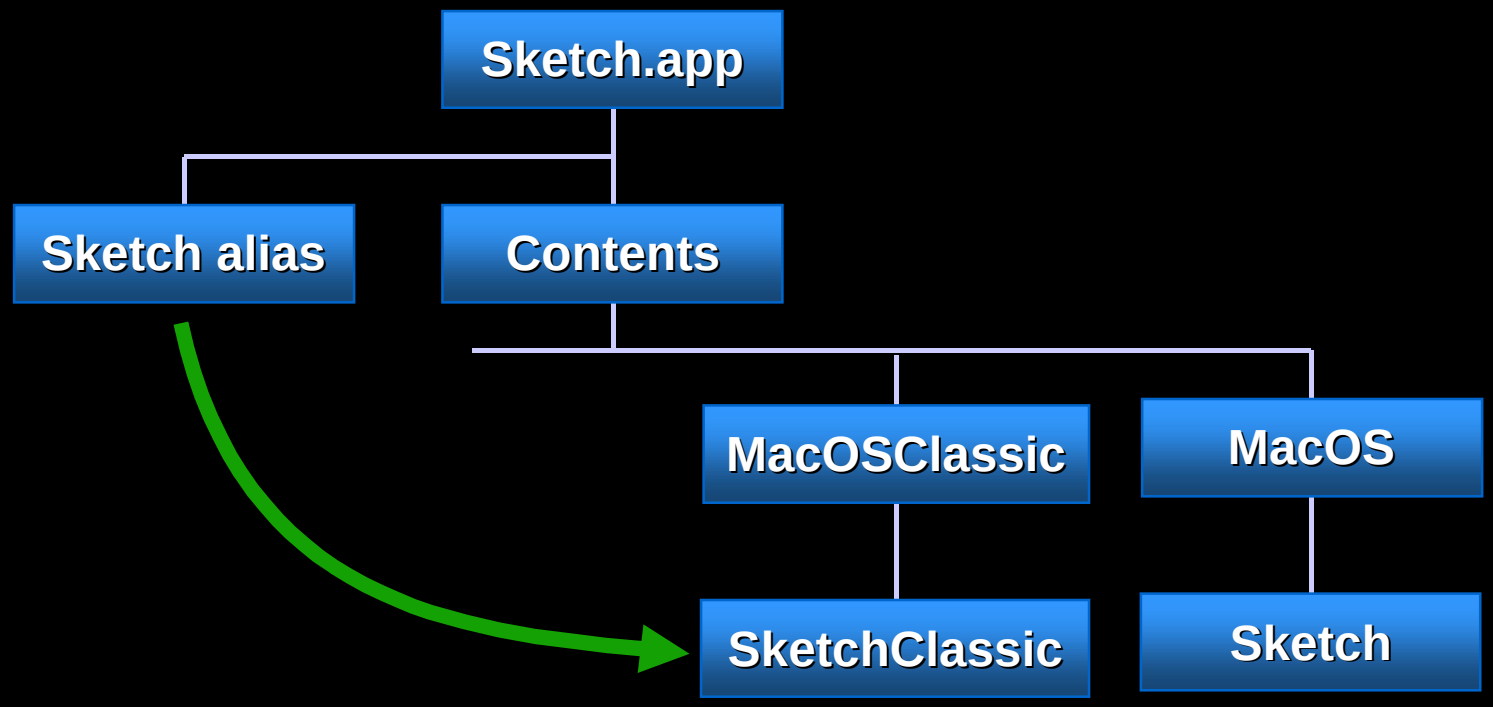
Application Packages for Mac OS 9



Application Packages for Mac OS 9



Application Packages for Mac OS 9



Demo

What Makes a Folder a Package?

Either…

- Has catalog info “package” bit set

What Makes a Folder a Package?

Either…

- Has catalog info “package” bit set

or…

- Has an extension *and* a **PkgInfo** file

What Makes a Folder a Package?

Either…

- Has catalog info “package” bit set

or…

- Has an extension *and* a **PkgInfo** file

or…

- Has an extension that is claimed as a package by a known application

The Property List

- Tells the system all about your application
  - Executable name
  - Version
  - Document types handled
  - URL schemes handled
  - *Extensible…*
- *All* bundles have a property list

The Property List

- Supercedes **vers**, **BNDL**, **FREF**, **open**, and **kind** resources
- XML format
- A dictionary with standard keys
- Some values should be localized in **InfoPlist.strings**

What? Your App Isn't Packaged?

- Add a **plst (0)** resource to your binary
- Paste the **Info.plist** bytes into it
- Icon file references become **icns** resource IDs
- A property list is the only way to
  - Get pretty icons
  - Claim extensions

Property List Keys

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Bundle Name

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Unique Identifier

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	**com.apple.sketch**
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Package Type

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	**APPL**
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Signature

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Version

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Icon File

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	Sketch

Executable File Name

Key	Value
CFBundleName	Sketch
CFBundleIdentifier	com.apple.sketch
CFBundlePackageType	APPL
CFBundleSignature	sktc
CFBundleVersion	10.0
CFBundleIconFile	Sketch.icns
CFBundleExecutable	**Sketch**

Document and URL Types

Key	Value
CFBundleDocumentTypes	Array of type info
CFBundleURLTypes	Array of URL info

Document Type Keys

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	sktc
CFBundleTypeExtensions	sketch, draw2

Type Name

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	sktc
CFBundleTypeExtensions	sketch, draw2

Type Icon File

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	sktc
CFBundleTypeExtensions	sketch, draw2

Type Role

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	sktc
CFBundleTypeExtensions	sketch, draw2

Type OSTypes Array

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	**sktc**
CFBundleTypeExtensions	sketch, draw2

Type Extensions Array

Key	Value
CFBundleTypeName	SketchDoc
CFBundleTypeIconFile	SketchDoc.icns
CFBundleTypeRole	Editor
CFBundleTypeOSTypes	sktc
CFBundleTypeExtensions	sketch, draw2

Demo

Application Packages Summary

- Packages offer great advantages over single-file applications
  - Simple user presentation
  - Globalized, not localized
  - Extensible
- The property list is the key

Document Binding

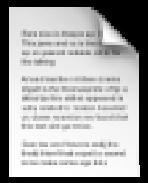
Document Binding

What is it?

- Document binding is how the system associates documents with applications that can open them
- Makes double-click work in the Finder
- Also refers to URL scheme binding

Binding 1984

Paradise Island

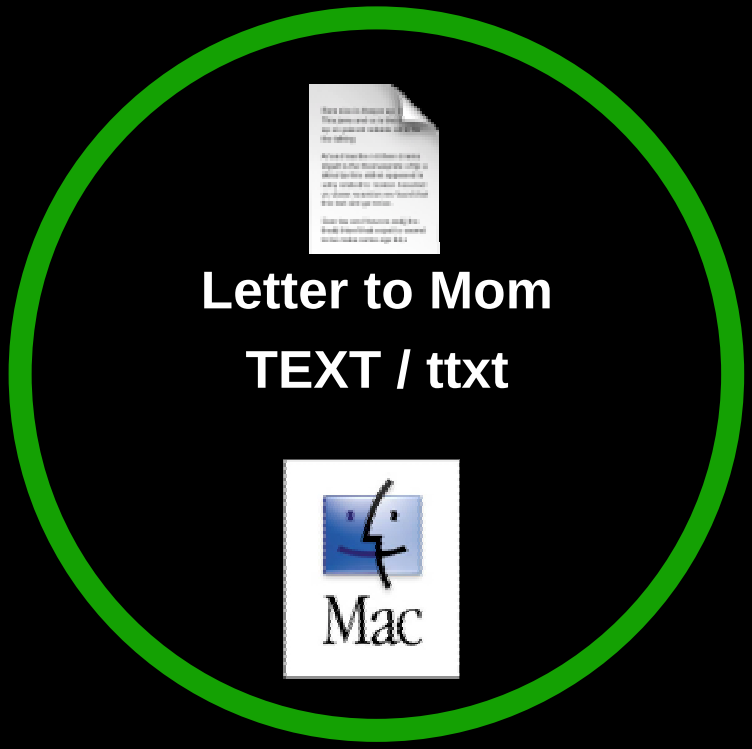


Letter to Mom

TEXT / ttxt

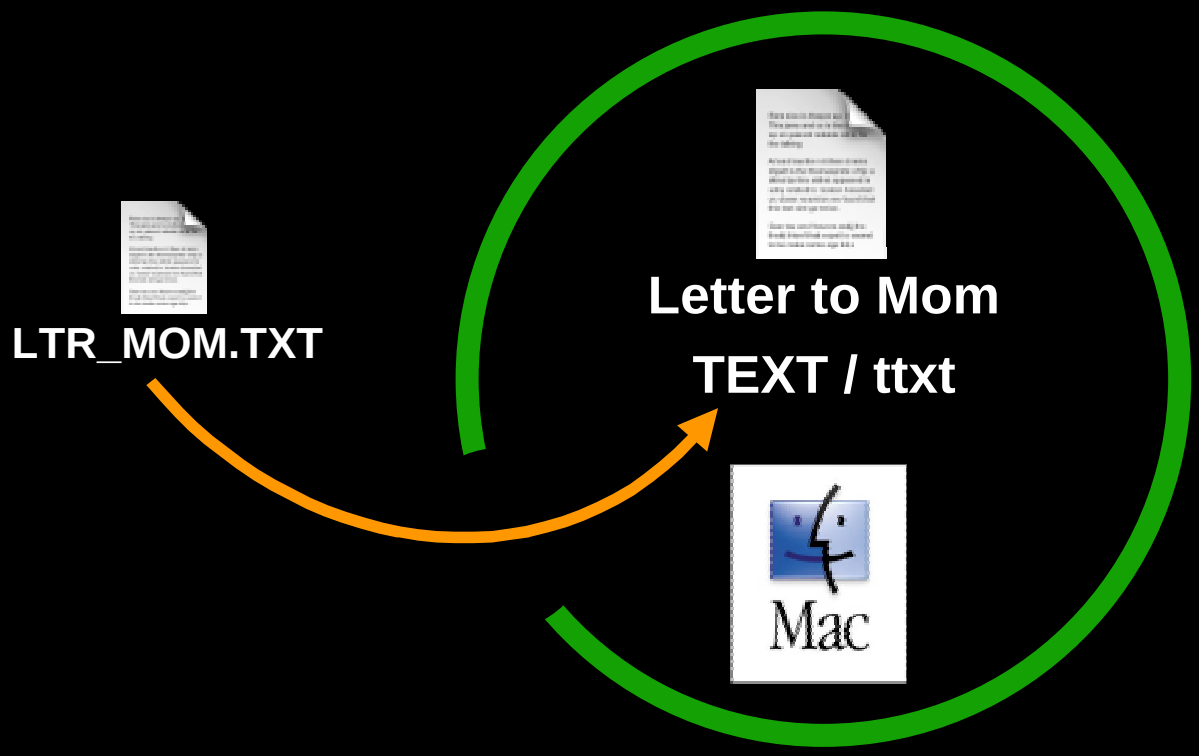
Binding 2001

Gatekeepers…



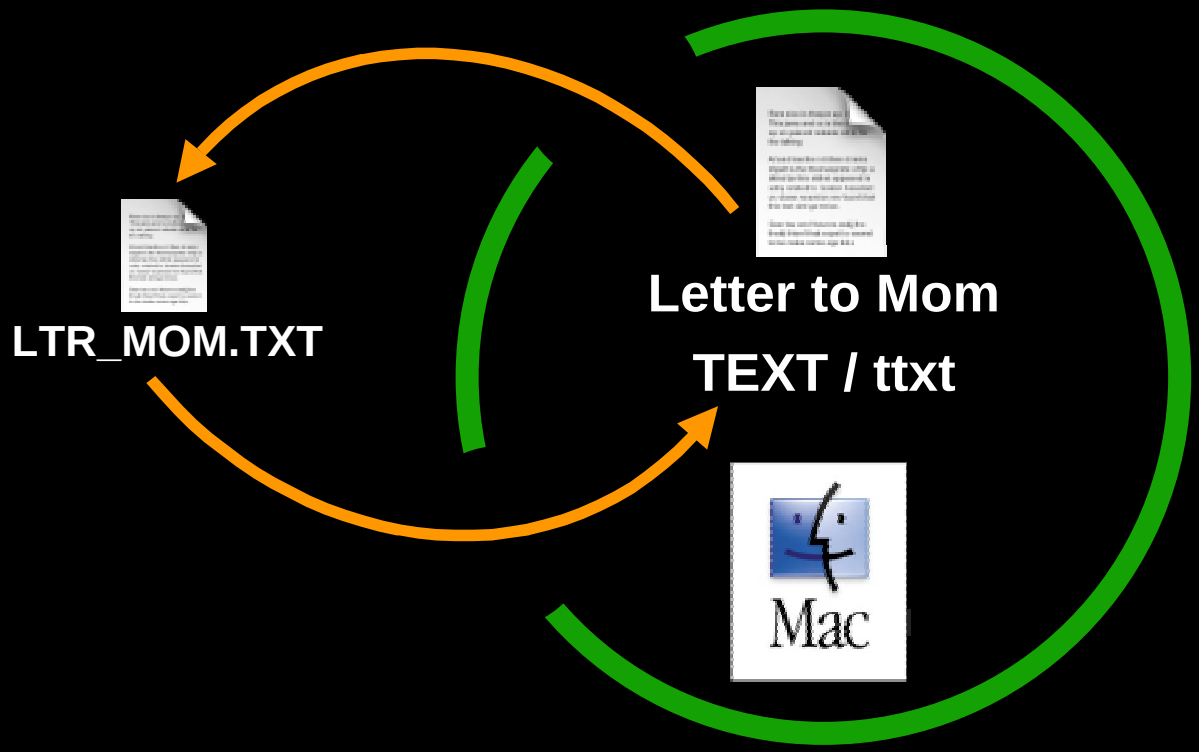
Binding 2001

Gatekeepers…



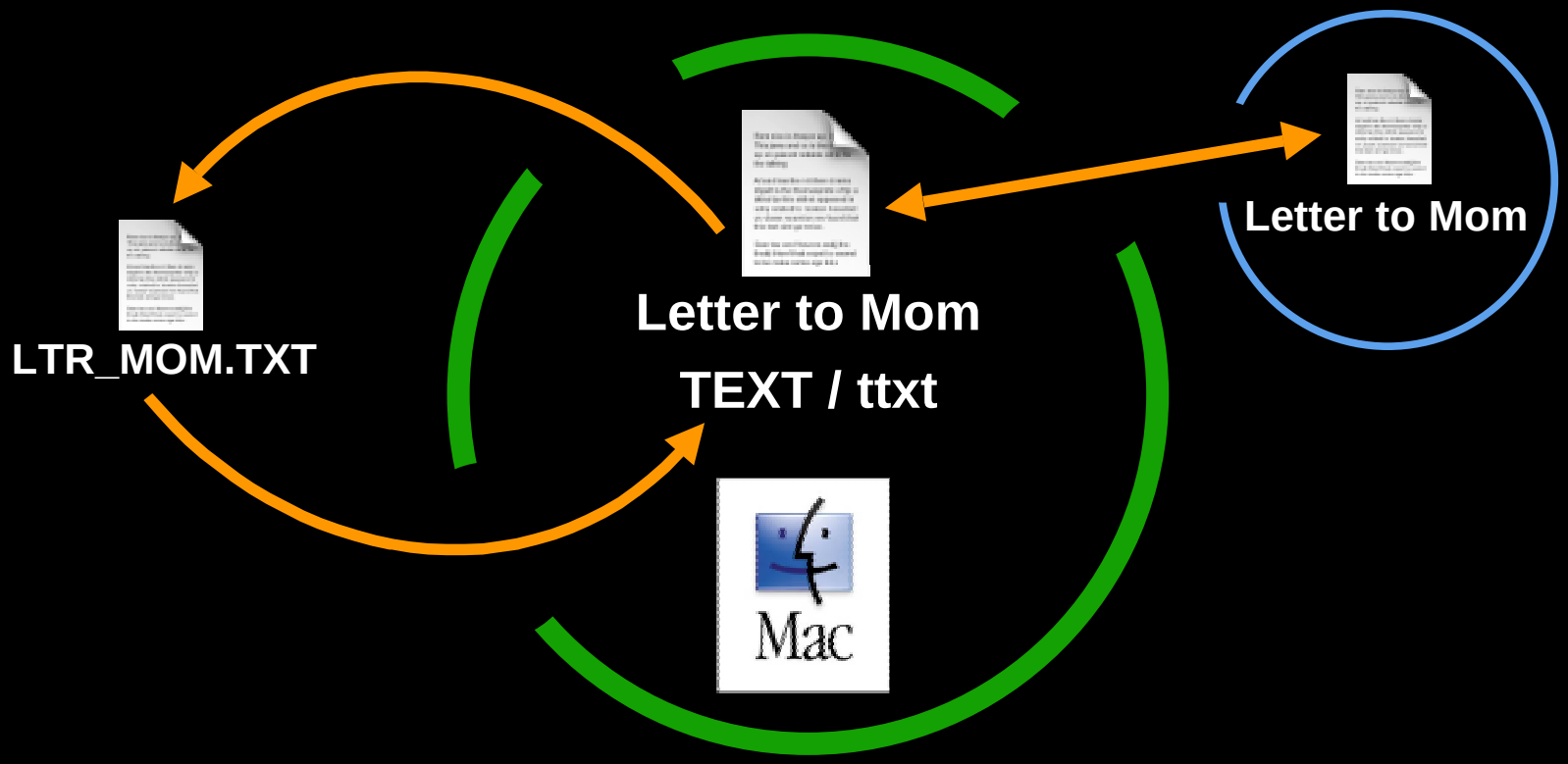
Binding 2001

Gatekeepers…



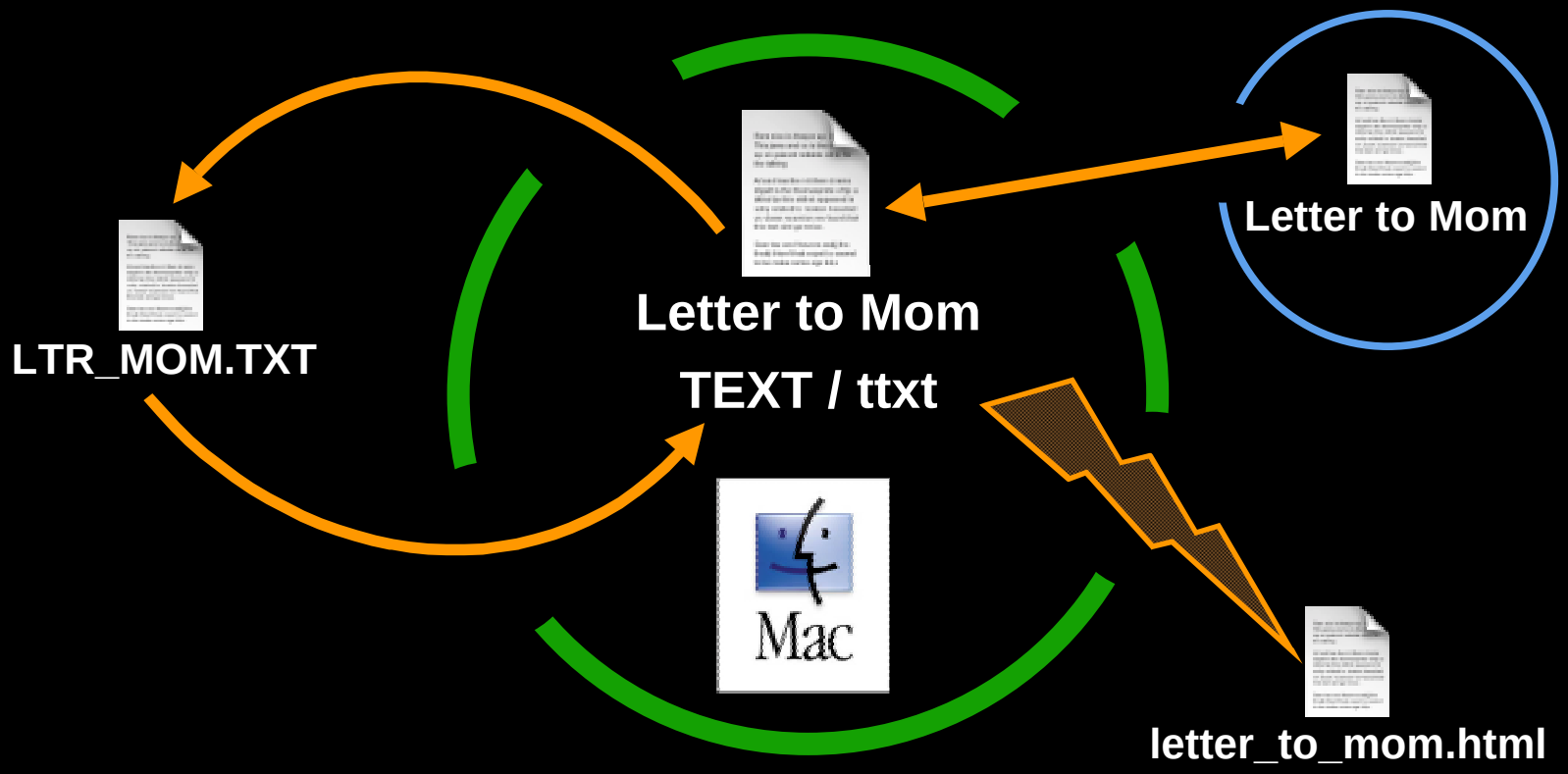
Binding 2001

Gatekeepers…

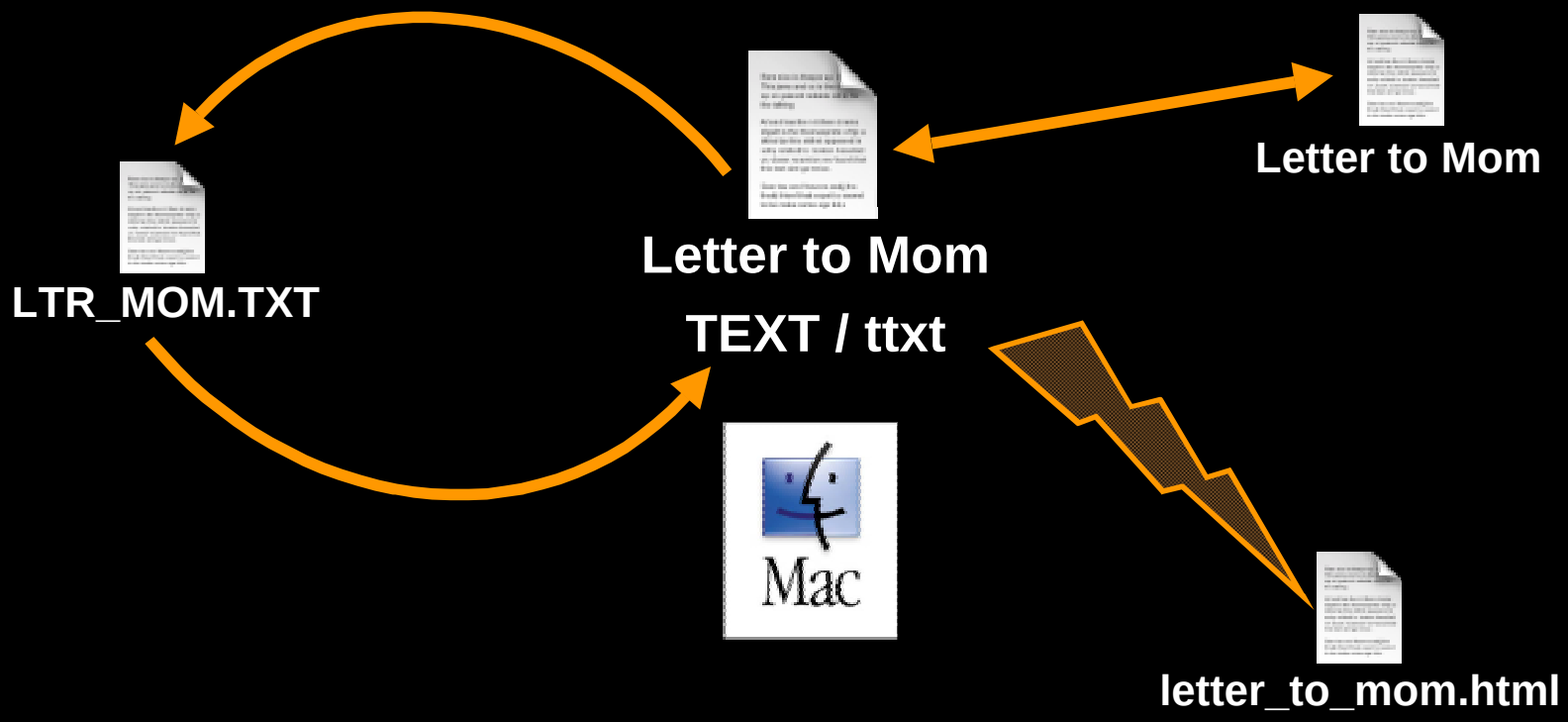


Binding 2001

Gatekeepers… with too many gates



Binding 2001



What Is the Answer?

What Is the Answer?

Assimilate Microsoft

What Is the Answer?

What Is the Answer?

Remodulate Shields

What Is the Answer?

What Is the Answer?

Recognize Extensions

Binding in Mac OS X

Features

- The goal is to always choose the “right” application when opening a document
- All available type information is recognized
- User has more control

What Is an Extension?

- Characters following the last dot
- Character set: **a-z 0-9 $ % _ ~**
- No spaces!
- Cannot be *only* digits
- Always matched case-insensitive

Examples

- Valid extensions:
  - **Doc.surfpaint**
  - **Doc.SurfPaint**
  - **ListView.c**
  - **Dido-Thanx.mp3**
- Not valid extensions:
  - **SurfPaint 1.0**
  - **SurfPaint 1.0 b2**
  - **Resume 05.23.01**

Binding in Mac OS X

What are the binding rules?

Binding in Mac OS X

What are the binding rules?

- User-override per-document binding

Binding in Mac OS X

What are the binding rules?

- User-override per-document binding
- Creator binding, best match

Binding in Mac OS X

What are the binding rules?

- User-override per-document binding
- Creator binding, best match
- Extension binding, best match

Binding in Mac OS X

What are the binding rules?

- User-override per-document binding
- Creator binding, best match
- Extension binding, best match
- OSType binding, best match

Creator Binding

Finding the best match

- Native app beats Classic app
- Latest version wins
- Latest modification date wins

OSType or Extension Binding

Finding the best match

- User's preferred app for type
- Editor beats viewer
- Native app beats Classic app

Guidelines for New Documents

What metadata should I set?

- To maintain binding for Mac OS 8/9 users, set both type and creator
- For cross-platform document exchange, assign an extension
- To work well everywhere, set all three
- Don’t manufacture missing metadata

Guidelines for New Documents

How do I make the Finder update?

- Finder does *not* poll periodically
- Call **FNNotify** when you save a new file
- Carbon and Cocoa save panels do this for you
  - **NavCompleteSave**
  - **NSSavePanel**

Wildcard Document Types

- Used for drag-and-drop acceptance
- Not used for Finder “Open” command
- Claiming extension '*'
  - App can open any file having an extension
- Claiming type '****'
  - App can open *any* file

So How Do I Open a Document?

Use Launch Services

- Open documents and URLs
- Get binding-related information about
  - Files
  - Folders
  - Applications

Launch Services

Specifics

- Available only on Mac OS X
- Maintains the bindings database
- Supercedes but also maintains the Desktop DB
- Used extensively by the Finder

Launch Services

Opening documents and URLs

- Easy way
  - **LSOpenFSRef()**
  - **LSOpenCFURLRef()**
- More control
  - **LSOpenFromRefSpec()**
  - **LSOpenFromURLSpec()**

Launch Services

Getting file or folder information

- Is this item a package, volume, or application?
- Is this item a classic or native application?
- What is this item's extension?
  - **LSCopyItemInfoForRef()**
  - **LSCopyItemInfoForURL()**

Launch Services

Getting application binding info

- Get best binding for a specific document or URL
  - **LSGetApplicationForItem()**
  - **LSGetApplicationForURL()**
- Get best binding for a given creator, type or extension
  - **LSGetApplicationForInfo()**

Launch Services

Getting application binding info

- Find an application matching any combo of application name, creator or bundle identifier
  - **LSFindApplicationForInfo()**

Launch Services

Checking drag and drop acceptance

- Can one item be dropped on another?
  - **LSCanRefAcceptItem()**
  - **LSCanURLAcceptURL()**

Launch Services

Kind Strings

- Get the localized kind string for a document, based on current binding
  - **LSCopyKindStringForRef()**
  - **LSCopyKindStringForURL()**

Launch Services

Housekeeping

- These functions do not need to be called
  - **LSInit()**
  - **LSTerm()**

How Are Bindings Established?

- The Finder updates the bindings database
- Well known places are scanned at login
  - Main Applications folder
  - User's Applications folder
- Other applications are added to the database when launched from the Finder

How Do I Add My Application?

Tell the Finder about it

- Create a kAESync AppleEvent
- Add an alias to your installed application
- Send the event to the Finder
- Finder adds your application to the bindings database

Installation Scenarios

- Packages work great for drag-and-drop installs
- For more formal installs:
  - Copy over the new application
  - Don't force a restart or logout
  - Send the kAESync event

Document Binding Summary

- All available type information is recognized
- User has more control
- Best practice is to set type, creator *and* extension for new documents
- Use Launch Services to mirror Finder actions

Roadmap

129 Mac OS X Application Localization	Room C **Thurs., 3:30pm**
130 Designing and Using Aqua Icons	Hall 2 **Fri., 10:30am**
138 Core Foundation Overview	Civic **Fri., 9:00am**
701 Project Builder Introduction	Hall 2 **Tues., 2:00pm**
703 Project Builder In-Depth	Room A2 **Wed., 9:00am**
Star Trek Voyager Final Episode	Hall 2 **Wed., 8:30pm**

Roadmap

119 Carbon Controls & Appearance I	Hall 2 **Thurs., 9:00am**
120 Carbon Controls & Appearance II	Hall 2 **Thurs., 10:30am**
125 Apple Help	Room J2 **Thurs., 2:00pm**
702 Interface Builder	Hall 2 **Tues., 3:30pm**
FF103 Feedback Forum: Aqua	Room J1 **Wed., 3:30pm**
FF114 Feedback Forum: High Level Toolbox	Room C **Fri., 3:30pm**

Q&A

John Geleynse
User Experience Technology Manager
Apple Worldwide Developer Relations
geleynse@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001